SIAM J. SCI. COMPUT.
Vol. 36, No. 3, pp. A1339–A1358

APPROXIMATE BAYESIAN COMPUTATION BY SUBSET SIMULATION*

MANUEL CHIACHIO†, JAMES L. BECK‡, JUAN CHIACHIO†, AND GUILLERMO RUS†

Abstract. A new approximate Bayesian computation (ABC) algorithm for Bayesian updating of model parameters is proposed in this paper, which combines the ABC principles with the technique of *subset simulation* for efficient rare-event simulation, first developed in S. K. Au and J. L. Beck [*Probabilistic Engrg. Mech.*, 16 (2001), pp. 263–277]. It has been named ABC-SubSim. The idea is to choose the nested decreasing sequence of regions in subset simulation as the regions that correspond to increasingly closer approximations of the actual data vector in observation space. The efficiency of the algorithm is demonstrated in two examples that illustrate some of the challenges faced in real-world applications of ABC. We show that the proposed algorithm outperforms other recent sequential ABC algorithms in terms of computational efficiency while achieving the same, or better, measure of accuracy in the posterior distribution. We also show that ABC-SubSim readily provides an estimate of the evidence (marginal likelihood) for posterior model class assessment, as a by-product.

Key words. approximate Bayesian computation, subset simulation, Bayesian inverse problem

AMS subject classifications. 65D05, 41A63, 41A10

DOI. 10.1137/130932831

1. Introduction.

The main goal of Bayesian statistics is to update a priori information about the parameter of interest $\theta \in \mathbf{\Theta} \subset \mathbb{R}^d$ for a parameterized model class $\mathcal{M}$, based on the information contained in a set of data which we express as a vector $y \in \mathcal{D} \subset \mathbb{R}^\ell$, where $\mathcal{D}$ is the *observation space*, the region in $\mathbb{R}^\ell$ of all possible observational outcomes according to the model class. As a part of the model class $\mathcal{M}$, we choose a prior probability density function (PDF) $p(\theta|\mathcal{M})$ over the parameter space and we also derive $p(y|\theta,\mathcal{M})$, the likelihood function of θ, from the stochastic forward model $p(x|\theta,\mathcal{M})$ of the model class $\mathcal{M}$ [6]. Bayes' theorem then yields the posterior PDF $p(\theta|y,\mathcal{M})$ of the model specified by θ as follows:

$$p(\theta|y,\mathcal{M}) = \frac{p(\theta|\mathcal{M})p(y|\theta,\mathcal{M})}{\int_{\mathbf{\Theta}} p(\theta|\mathcal{M})p(y|\theta,\mathcal{M})d\theta} \propto p(\theta|\mathcal{M})p(y|\theta,\mathcal{M}). \tag{1.1}$$

However, evaluation of the normalizing integral in the denominator is usually intractable except in some special cases. Also, there are situations where Bayesian analysis is conducted with a likelihood function that is not completely known or is difficult to obtain, perhaps because it requires the evaluation of an intractable multi-dimensional integral over a latent vector, such as in hidden Markov models or dynamic state-space models, or because the normalization in the likelihood over the observation space $\mathcal{D}$ involves an intractable integral parameterized by θ [22]. Approximate

*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 13, 2013; accepted for publication (in revised form) April 8, 2014; published electronically June 26, 2014. This work was supported by the Spanish Ministry of Economy for project DPI2010-17065 and the European Union for the "Programa Operativo FEDER de Andalucía 2007-2013" for project GGI3000IDIB.

http://www.siam.org/journals/sisc/36-3/93283.html

†Department of Structural Mechanics and Hydraulic Engineering, University of Granada. Campus de Fuentenueva s/n, 18071 Granada, Spain (mchiachio@ugr.es, jchiachio@ugr.es, grus@ugr.es). The first author was supported by the Education Ministry of Spain for the FPU grants AP2009-4641, AP2009-2390.

‡Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125 (jimbeck@caltech.edu).

Bayesian computation (ABC) algorithms were conceived with the aim of evaluating the posterior density in those cases where the likelihood function is intractable [33, 26], although it also avoids the problem of the intractable integral in (1.1). In the literature, these classes of algorithms are also called *likelihood-free computation algorithms*, which refers to their main aim of circumventing the explicit evaluation of the likelihood by using a simulation-based approach. In this introductory section, we briefly summarize the body of ABC literature with a brief description of the main concepts and algorithms that we will need in the subsequent sections.

Let $x \in \mathcal{D} \subset \mathbb{R}^\ell$ denote a simulated dataset from $p(\cdot|\theta, \mathcal{M})$, the forward model of model class $\mathcal{M}$. An ABC algorithm aims at evaluating the posterior $p(\theta|y, \mathcal{M}) \propto p(y|\theta, \mathcal{M})p(\theta|\mathcal{M})$ by applying Bayes' theorem to the pair (θ, x):

$$p(\theta, x|y) \propto p(y|x, \theta)p(x|\theta)p(\theta). \tag{1.2}$$

In the last equation, the conditioning on model class $\mathcal{M}$ has been omitted for clarity, given that the theory is valid for any specific model class. The function $p(y|x, \theta)$ gives higher weights for the posterior in those regions where x is close to y. The basic form of the algorithm to sample from the posterior given by (1.2) is a rejection algorithm that consists of generating jointly $\theta \sim p(\theta)$ and $x \sim p(x|\theta)$ and accepting them conditional on fulfilling the equality $x = y$. Of course, obtaining sample $x = y$ is unlikely in most applications, and it is only feasible if $\mathcal{D}$ consists of a finite set of values rather than a region in $\mathbb{R}^\ell$. Hence two main approximations have been conceived in ABC theory to address this difficulty [24]: (a) replace the equality $x = y$ by the approximation $x \approx y$ and introduce a tolerance parameter ϵ that accounts for how close they are through some type of metric ρ, and (b) introduce a low-dimensional vector of summary statistics $\eta(\cdot)$ that permits a comparison of the closeness of x and y in a weak manner. Through this approach, the posterior $p(\theta, x|y)$ in (1.2) is approximated by $p_\epsilon(\theta, x|y)$, which assigns higher probability density to those values of $(\theta, x) \in \boldsymbol{\Theta} \times \mathcal{D}$ that satisfy the condition $\rho\big(\eta(x), \eta(y)\big) \leqslant \epsilon$.

The standard version of the ABC algorithm takes the approximate likelihood[1] $P_\epsilon(y|\theta, x) = P(x \in \mathcal{N}_\epsilon(y)|x)$, where $\mathcal{N}_\epsilon(y) = \{x \in \mathcal{D} : \rho(\eta(x), \eta(y)) \leqslant \epsilon\}$. From Bayes' theorem, the approximate posterior $p_\epsilon(\theta, x|y) = p(\theta, x|x \in \mathcal{N}_\epsilon(y))$ is given by

$$p_\epsilon(\theta, x|y) \propto P(x \in \mathcal{N}_\epsilon(y)|x)p(x|\theta)p(\theta), \tag{1.3}$$

where $P(x \in \mathcal{N}_\epsilon(y)|x) = \mathbb{I}_{\mathcal{N}_{\epsilon(y)}}(x)$, an indicator function for the set $\mathcal{N}_\epsilon(y)$ that assigns a value of 1 when $\rho(\eta(x), \eta(y)) \leqslant \epsilon$ and 0 otherwise. So the output of the ABC algorithm corresponds to samples from the joint PDF,

$$p_\epsilon(\theta, x|y) \propto p(x|\theta)p(\theta)\mathbb{I}_{\mathcal{N}_{\epsilon(y)}}(x), \tag{1.4}$$

with ultimate interest typically being in the marginal approximate posterior:

$$p_\epsilon(\theta|y) \propto p(\theta) \int_{\mathcal{D}} p(x|\theta)\mathbb{I}_{\mathcal{N}_{\epsilon(y)}}(x)dx = P(x \in \mathcal{N}_\epsilon(y)|\theta)p(\theta). \tag{1.5}$$

This integration need not be done explicitly since samples from this marginal PDF are obtained by taking the θ component of samples from the joint PDF in (1.4) [27]. Notice that the quality of the posterior approximation in (1.4) and (1.5) depends on a suitable selection of the metric ρ, the tolerance parameter ϵ, and, of special importance, the summary statistic $\eta(\cdot)$ [14]. A pseudocode to generate N samples by the standard version of ABC algorithm is given in Algorithm 1.

[1] In what follows, we use $P(\cdot)$ to denote probability whereas a PDF is expressed as $p(\cdot)$.

ALGORITHM 1. STANDARD ABC.

for $n = 1$ to N **do**
  repeat
    1. Simulate θ' from $p(\theta)$
    2. Generate $x' \sim p(x|\theta')$
  until $\rho\big(\eta(x'), \eta(y)\big) \leqslant \epsilon$
  Accept (θ', x')
end for

The choice of tolerance parameter ϵ is basically a matter of the amount of computational effort that the user wishes to expend, but a possible guiding principle is described later at the end of section 3.1.2. For ϵ sufficiently small ($\epsilon \to 0$), $\eta(x) \to \eta(y)$, and so all accepted samples corresponding to (1.5) come from the closest approximation to the required posterior density $p(\theta|y)$, where the exactness is achieved when $\eta(\cdot)$ is a sufficient statistic. This desirable fact is at the expense of a high computational effort (usually prohibitive) to get $\eta(x) = \eta(y)$ under the model $p(x|\theta)$. On the contrary, as $\epsilon \to \infty$, all accepted observations come from the prior. So, the choice of ϵ reflects a trade-off between computability and accuracy.

Several computational improvements have been proposed addressing this trade-off. In those cases where the probability content of the posterior is concentrated over a small region in relation to a diffuse prior, the use of Markov chain Monte Carlo methods (MCMC) [17, 25, 16] has been demonstrated to be efficient [24]. In fact, the use of a proposal PDF $q(\cdot|\cdot)$ over the parameter space allows a new parameter to be proposed based on a previously accepted one, targeting the stationary distribution $p_\epsilon(\theta|y)$. The resulting algorithm, commonly called ABC-MCMC, is similar to the standard one (Algorithm 1) with the main exception being the acceptance probability, which in this case is influenced by the MCMC acceptance probability as follows.

ALGORITHM 2. ABC-MCMC.

1. Initialize $(\theta^{(0)}, x^{(0)})$ from $p_\epsilon(\theta, x|y)$, e.g., use Algorithm 1.
for $n = 1$ to N **do**
  2. Generate $\theta' \sim q(\theta|\theta^{(n-1)})$ and $x' \sim p(x|\theta')$
  3. Accept (θ', x') as $(\theta^{(n)}, x^{(n)})$ with probability:
  $\alpha = \min\left\{1, \frac{P_\epsilon(y|x',\theta')p(\theta')q(\theta^{(n-1)}|\theta')}{P_\epsilon(y|x^{(n-1)},\theta^{(n-1)})p(\theta^{(n-1)})q(\theta'|\theta^{(n-1)})}\right\}$
  else set $(\theta^{(n)}, x^{(n)}) = (\theta^{(n-1)}, x^{(n-1)})$
end for

When $P_\epsilon(y|x,\theta) = \mathbb{I}_{\mathcal{N}_{\epsilon(y)}}(x)$, as in our case, the acceptance probability α is decomposed into the product of the MCMC acceptance probability and the indicator function:

$$\alpha = \min\left\{1, \frac{p(\theta')q(\theta^{(n-1)}|\theta')}{p(\theta^{(n-1)})q(\theta'|\theta^{(n-1)})}\right\}\mathbb{I}_{\mathcal{N}_{\epsilon(y)}}(x'). \tag{1.6}$$

In this case, step 3 is performed only if $x' \in \mathcal{N}_\epsilon(y)$. The efficiency of this algorithm is improved with respect to the standard ABC algorithm, but (1.6) clearly shows that the dependence upon ϵ in the indicator function may lead to an inefficient algorithm for a good approximation of the true posterior. In fact, given that α can only be nonzero if the event $\rho(\eta(x^{'}), \eta(y)) \leqslant \epsilon$ occurs, the chain may persist in distributional tails for long periods of time if ϵ is sufficiently small, due to the acceptance probability being zero in step 3 of Algorithm 2.

TABLE 1

Bibliography synoptic table about ABC with sequential algorithms. Papers ordered by increasing date of publication. PRC: partial rejection control, SMC: sequential Monte Carlo, PT: parallel tempering, PMC: population Monte Carlo.

Paper	Algorithm	Year	Notes
Sisson, Fan, and Tanaka [31]	ABC-PRC	2007	Requires forward and a backward kernels to perturb the particles. Uses an SMC sampler. Induces bias.
Toni et al. [34]	ABC-SMC	2009	Does not require resampling steps in [31]. Based on sequential importance sampling. Induces bias.
Sisson, Fan, and Tanaka [32]	ABC-PRC	2009	This version incorporates an improved weight updating function. Outperforms original in [31].
Beaumont et al. [5]	ABC-PMC	2009	Does not require a backward kernel as in the preceding works [31, 32].
Baragatti, Grimaud, and Pommeret [4]	ABC-PT	2011	Based on MCMC with exchange moves between chains. Capacity to exit from distribution tails.
Drovandi and Pettitt [13]	Adaptive ABC-SMC	2011	Outperforms original in [34]. Automatic determination of the tolerance sequence $\epsilon_j, j = \{1, \ldots, m\}$ and the proposal distribution of the MCMC kernel.
Del Moral, Doucet, and Jasra [11]	Adaptive ABC-SMC	2012	More efficient than ABC-SMC [34, 13]. Automatic determination of the tolerance sequence $\epsilon_j, j = \{1, \ldots, m\}$.

Some modifications to the ABC-MCMC scheme have been proposed [8] that provide a moderate improvement in the simulation efficiency. See [30] for a complete tutorial about ABC-MCMC. More recently, to overcome this drawback associated with ABC-MCMC, a branch of computational techniques have emerged to obtain high accuracy ($\epsilon \to 0$) with a feasible computational burden by combining sequential sampling algorithms [10] adapted for ABC. These techniques share a common principle of achieving computational efficiency by learning about intermediate target distributions determined by a decreasing sequence of tolerance levels $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_m = \epsilon$, where the last is the desired tolerance ϵ. Table 1 lists the main contributions to the literature on this topic. However, more research is needed to perform posterior simulations in a more efficient manner.

In this paper we introduce a new sequential algorithm, called *ABC based on subset simulation* (ABC-SubSim), which combines the ABC principle with the technique of subset simulation [1, 2, 3] to achieve computational efficiency in a sequential way. The main idea is to link an ABC algorithm with a highly efficient rare-event sampler that draws conditional samples from a nested sequence of subdomains defined in an adaptive and automatic manner. ABC-SubSim can utilize many of the improvements proposed in the recent ABC literature because of the fact that the algorithm is focused on the core simulation engine.

The paper is organized as follows. Section 2 reviews the theory underlying subset simulation, and then the ABC-SubSim algorithm is introduced in section 3. The

efficiency of ABC-SubSim is illustrated in section 4 with two examples of dynamical models with synthetic data. In section 5, the performance of the algorithm is compared with some others in the recent ABC literature and the use of ABC-SubSim for posterior model class assessment is discussed. Section 6 provides concluding remarks.

2. Subset simulation method. Subset simulation is a simulation approach originally proposed to compute small failure probabilities encountered in reliability analysis of engineering systems (e.g., [1, 2, 9]). Strictly speaking, it is a method for efficiently generating conditional samples that correspond to specified levels of a performance function $g : \mathbb{R}^d \to \mathbb{R}$ in a progressive manner, converting a problem involving rare-event simulation into a sequence of problems involving more frequent events.

Let F be the failure region in the z-space, $z \in Z \subset \mathbb{R}^d$, corresponding to exceedance of the performance function above some specified threshold level b:

$$F = \{z \in Z : g(z) > b\}. \tag{2.1}$$

For simpler notation, we use $P(F) \equiv P(z \in F)$. Let us now assume that F is defined as the intersection of m regions $F = \bigcap_{j=1}^{m} F_j$, such that they are arranged as a nested sequence $F_1 \supset F_2 \cdots \supset F_{m-1} \supset F_m = F$, where $F_j = \{z \in Z : g(z) > b_j\}$ with $b_{j+1} > b_j$, such that $p(z|F_j) \propto p(z)\mathbb{I}_{F_j}(z)$, $j = 1, \ldots, m$. The term $p(z)$ denotes the probability model for z. When the event F_j holds, $\{F_{j-1}, \ldots, F_1\}$ also hold, and hence $P(F_j|F_{j-1}, \ldots, F_1) = P(F_j|F_{j-1})$, so it follows that

$$P(F) = P\left(\bigcap_{j=1}^{m} F_j\right) = P(F_1)\prod_{j=2}^{m} P(F_j|F_{j-1}), \tag{2.2}$$

where $P(F_j|F_{j-1}) \equiv P(z \in F_j|z \in F_{j-1})$ is the conditional failure probability at the $(j-1)$th conditional level. Notice that although the probability $P(F)$ can be relatively small, by choosing the intermediate regions appropriately, the conditional probabilities involved in (2.2) can be made large, thus avoiding simulation of rare events.

In the last equation, apart from $P(F_1)$, the remaining factors cannot be efficiently estimated by the standard Monte Carlo method because of the conditional sampling involved, especially at higher intermediate levels. Therefore, in subset simulation, only the first probability $P(F_1)$ is estimated by the standard Monte Carlo method:

$$P(F_1) \approx \bar{P}_1 = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}_{F_1}(z_0^{(n)}), \qquad z_0^{(n)} \overset{\text{i.i.d.}}{\sim} p(z_0). \tag{2.3}$$

When $j \geqslant 2$, sampling from the PDF $p(z_{j-1}|F_{j-1})$ can be achieved by using MCMC at the expense of generating N dependent samples, giving

$$P(F_j|F_{j-1}) \approx \bar{P}_j = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}_{F_j}(z_{j-1}^{(n)}), \qquad z_{j-1}^{(n)} \sim p(z_{j-1}|F_{j-1}), \tag{2.4}$$

where $\mathbb{I}_{F_j}(z_{j-1}^{(n)})$ is the indicator function for the region $F_j, j = 1, \ldots, m$, that assigns a value of 1 when $g(z_{j-1}^{(n)}) > b_j$, and 0 otherwise.

Observe that the Markov chain samples that are generated at the $(j-1)$th level which lie in F_j are distributed as $p(z|F_j)$ and thus they provide "seeds" for simulating

more samples according to $p(z|F_j)$ by using MCMC sampling with no burn-in required. As described further below, F_j is actually chosen adaptively based on the samples $\{z_{j-1}^{(n)}, n = 1, \ldots, N\}$ from $p(z|F_{j-1})$ in such a way that there are exactly NP_0 of these seed samples in F_j (so $\bar{P}_j = P_0$ in (2.4)). Then a further $(1/P_0 - 1)$ samples are generated from $p(z|F_j)$ by MCMC starting at each seed, giving a total of N samples in F_j. Repeating this process, we can compute the conditional probabilities of the higher conditional levels until the final region $F_m = F$ has been reached.

To draw samples from the target PDF $p(z|F_j)$ using the Metropolis algorithm, a suitable proposal PDF must be chosen. In the original version of subset simulation [1], a modified Metropolis algorithm (MMA) was proposed that works well even in very high dimensions (e.g., 10^3-10^4), because the original algorithm fails in this case. (Essentially all candidate samples from the proposal PDF are rejected; see the analysis in [1].) In MMA, a univariate proposal PDF is chosen for each component of the parameter vector and each component candidate is accepted or rejected separately, instead of drawing a full parameter vector candidate from a multidimensional PDF as in the original algorithm. Later in [2], grouping of the parameters was considered when constructing a proposal PDF to allow for the case where small groups of components in the parameter vector are highly correlated when conditioned on any F_j. An appropriate choice for the proposal PDF for ABC-SubSim is introduced in the next section.

It is important to remark that in subset simulation, an inadequate choice of the b_j-sequence may lead to the conditional probability $P(F_j|F_{j-1})$ being very small (if the difference $b_j - b_{j-1}$ is too large), which will lead to a rare-event simulation problem. If, on the contrary, the intermediate threshold values were chosen too close so that the conditional failure probabilities were very high, the algorithm would take a large total number of simulation levels m (and hence large computational effort) to progress to the target region of interest, F. A rational choice that strikes a balance between these two extremes is to choose the b_j-sequence adaptively [1], so that the estimated conditional probabilities are equal to a fixed value P_0 (e.g., $P_0 = 0.2$). For convenience, P_0 is chosen so that NP_0 and $1/P_0$ are positive integers. For a specified value of P_0, the intermediate threshold value b_j defining F_j is obtained in an automated manner as the $[(1 - P_0)N]$th largest value among the values $g(z_{j-1}^{(n)})$, $n = 1, \ldots, N$, so that the sample estimate of $P(F_j|F_{j-1})$ in (2.4) is equal to P_0.

3. Subset simulation for ABC. Here we exploit subset simulation as an efficient sampler for the inference of rare events by just specializing the subset simulation method described in section 2 to ABC. To this end, let us define z as $z = (\theta, x) \in Z = \boldsymbol{\Theta} \times \mathcal{D} \subset \mathbb{R}^{d+\ell}$, so that $p(z) = p(x|\theta)p(\theta)$. Let also F_j in section 2 be replaced by a nested sequence of regions D_j, $j = 1 \ldots, m$, in Z defined by

$$D_j = \Big\{ z \in Z : x \in \mathcal{N}_{\epsilon_j}(y) \Big\} \equiv \Big\{ (\theta, x) : \rho\big(\eta(x), \eta(y)\big) \leqslant \epsilon_j \Big\} \tag{3.1}$$

with $D_j \subset \boldsymbol{\Theta} \times \mathcal{D}$ and ρ a metric on the set $\{\eta(x) : x \in \mathcal{D}\}$. The sequence of tolerances $\epsilon_1, \epsilon_2, \ldots, \epsilon_m$ with $\epsilon_{j+1} < \epsilon_j$ will be chosen adaptively as described in section 2, where the number of levels m is chosen so that $\epsilon_m \leqslant \epsilon$, a specified tolerance.

As stated by (1.4), an ABC algorithm aims at evaluating the sequence of intermediate posteriors $p(\theta, x|D_j)$, $j = 1, \ldots, m$, where by Bayes' theorem

$$p(\theta, x|D_j) = \frac{P(D_j|\theta, x)p(x|\theta)p(\theta)}{P(D_j)} \propto \mathbb{I}_{D_j}(\theta, x)p(x|\theta)p(\theta). \tag{3.2}$$

Here, $\mathbb{I}_{D_j}(\theta, x)$ is the indicator function for the set D_j. Notice that when $\epsilon \to 0$, D_m represents a small closed region in Z and hence $P(D_m)$ will be very small under the model $p(\theta, x) = p(x|\theta)p(\theta)$. In this situation, using MCMC sampling directly is not efficient due to difficulties in initializing the chain and in achieving convergence to the stationary distribution, as was described in section 1 for ABC-MCMC. This is the point at which we exploit the efficiency of subset simulation for ABC, given that such a small probability $P(D_m)$ is converted into a sequence of larger conditional probabilities, as stated in (2.2), (2.3), and (2.4).

3.1. The ABC-SubSim algorithm. Algorithm 3 provides a pseudocode implementation of ABC-SubSim that is intended to be sufficient for most situations. The algorithm is implemented such that a maximum allowable number of simulation levels (m) is considered in case the specified ϵ is too small. The choice of ϵ is discussed at the end of section 3.1.2.

ALGORITHM 3. PSEUDOCODE IMPLEMENTATION FOR ABC-SUBSIM.

Inputs:

$P_0 \in [0, 1]$ {gives percentile selection, chosen so $NP_0, {}^1/_{P_0} \in \mathbb{Z}^+$; $P_0 = 0.2$ is recommended}.

N, {number of samples per intermediate level}; m, {maximum number of simulation levels allowed}

Algorithm:

Sample $\left[\left(\theta_0^{(1)}, x_0^{(1)}\right), \ldots, \left(\theta_0^{(n)}, x_0^{(n)}\right), \ldots, \left(\theta_0^{(N)}, x_0^{(N)}\right)\right]$, where $(\theta, x) \sim p(\theta)p(x|\theta)$

for $j : 1, \ldots, m$ **do**

  for $n : 1, \ldots, N$ **do**

    Evaluate $\rho_j^{(n)} = \rho\left(\eta(x_{j-1}^{(n)}), \eta(y)\right)$

  end for

  Renumber $\left[\left(\theta_{j-1}^{(n)}, x_{j-1}^{(n)}\right), n : 1, \ldots, N\right]$ so that $\rho_j^{(1)} \leqslant \rho_j^{(2)} \leqslant \cdots \rho_j^{(N)}$

  Fix $\epsilon_j = \frac{1}{2}\left(\rho_j^{(NP_0)} + \rho_j^{(NP_0+1)}\right)$

  for $k = 1, \ldots, NP_0$ **do**

    Select as a seed $\left(\theta_j^{(k),1}, x_j^{(k),1}\right) = \left(\theta_{j-1}^{(k)}, x_{j-1}^{(k)}\right) \sim p\left(\theta, x|(\theta, x) \in D_j\right)$

    Run MMA [1] to generate $1/P_0$ states of a Markov chain lying in D_j (3.1): $\left[\left(\theta_j^{(k),1}, x_j^{(k),1}\right), \ldots, \left(\theta_j^{(k),1/P_0}, x_j^{(k),1/P_0}\right)\right]$

  end for

  Renumber $\left[\left(\theta_j^{(k),i}, x_j^{(k),i}\right) : k = 1, \ldots, NP_0;\ i = 1, \ldots, 1/P_0\right]$ as $\left[\left(\theta_j^{(1)}, x_j^{(1)}\right), \ldots, \left(\theta_j^{(N)}, x_j^{(N)}\right)\right]$

  if $\epsilon_j \leqslant \epsilon$ **then**

    End algorithm

  end if

end for

3.1.1. Choice of intermediate tolerance levels. In Algorithm 3, the ϵ_j values are chosen adaptively as in subset simulation [1], so that the sample estimate $\bar{P}_j$ of $P(D_j|D_{j-1})$ satisfies $\bar{P}_j = P_0$. By this way, the intermediate tolerance value ϵ_j can be simply obtained as the $100P_0$ percentile of the set of distances $\rho(\eta(x_{j-1}^{(n)}), \eta(y)), n = 1, \ldots, N$, arranged in increasing order. Additionally, for convenience of implementation, we choose P_0 such that NP_0 and ${}^1/_{P_0}$ are integers, and so the size of the subset of samples generated in D_{j-1} that lie in D_j is known in advance

and equal to NP_0. These NP_0 samples in D_j are used as seeds for NP_0 Markov chains of length $1/P_0$, where the new $(1/P_0 - 1)$ samples in D_j in each chain are generated by MMA [1]. Hence the total number of samples of (θ, x) lying in D_j is N, but NP_0 of them were generated at the $(j-1)$th level. Because of the way the seeds are chosen, ABC-SubSim exhibits the benefits of *perfect sampling* [37, 27], which is an important feature to avoid wasting samples during a burn-in period, in contrast to ABC-MCMC.

3.1.2. Choosing ABC-SubSim control parameters. The important control parameters to be chosen in Algorithm 3 are P_0 and σ_j^2, the variance in the Gaussian proposal PDF in MMA at the jth level. In this section we make recommendations for the choice of these control parameters.

In the literature, the optimal variance of a local proposal PDF for an MCMC sampler has been studied due to its significant impact on the speed of convergence of the algorithm [15, 29]. ABC-SubSim has the novelty of incorporating the subset simulation procedure in the ABC algorithm, so we use the same optimal adaptive scaling strategy as in subset simulation. To avoid duplicating the literature for this technique but to confer a sufficient conceptual framework, the method for the optimal choice of the σ_j^2 is presented in a brief way. The reader is referred to the recent work of [37], where optimal scaling is addressed for subset simulation and a brief historical overview is also given on the topic.

Suppose that the reason for wanting to generate posterior samples is that we wish to calculate the posterior expectation of a quantity of interest which is a function $h : \theta \in \boldsymbol{\Theta} \to \mathbb{R}$. We consider the estimate of its expectation with respect to the samples generated in each of the jth levels:

$$\bar{h}_j = \mathbb{E}_{p_\epsilon(\theta|D_j)}[h(\theta)] \approx \frac{1}{N}\sum_{n=1}^{N} h(\theta_j^{(n)}), \tag{3.3}$$

where $\theta_j^{(n)}, n = 1, \ldots, N$ are dependent samples drawn from N_c Markov chains generated at the jth conditional level. An expression for the variance of the estimator can be written as follows [1]:

$$\mathrm{Var}(\bar{h}_j) = \frac{R_j^{(0)}}{N}(1 + \gamma_j) \tag{3.4}$$

with

$$\gamma_j = 2\sum_{\tau=1}^{N_s-1}\left(\frac{N_s - \tau}{N_s}\right)\frac{R_j^{(\tau)}}{R_j^{(0)}}. \tag{3.5}$$

In the last equation, $N_s = 1/P_0$ is the length of each of the Markov chains, which are considered probabilistically equivalent [1]. The term $R_j^{(\tau)}$ is the autocovariance of $h(\theta)$ at lag τ, $R_j^{(\tau)} = \mathbb{E}[h(\theta_j^{(1)})h(\theta_j^{(\tau+1)})] - \bar{h}_j^2$, which can be estimated using the Markov chain samples $\{\theta_j^{(k),i} : k = 1, \ldots, N_c;\ i = 1, \ldots, N_s\}$ as[2]

$$R_j^{(\tau)} \approx \tilde{R}_j^{(\tau)} = \left[\frac{1}{N - \tau N_c}\sum_{k=1}^{N_c}\sum_{i=1}^{N_s-\tau} h(\theta_j^{(k),i})h(\theta_j^{(k),\tau+i})\right] - \bar{h}_j^2, \tag{3.6}$$

where $N_c = NP_0$, so that $N = N_c N_s$.

[2]It is assumed for simplicity in the analysis that the samples generated by the different N_c chains are uncorrelated under the performance function h, although the samples are actually dependent because the seeds may be correlated. See further details in [1, section 6.2].

Given that the efficiency of the estimator $\bar{h}_j$ is reduced when γ_j is high, the optimal proposal variance σ_j^2 for simulation level jth is chosen adaptively by minimizing γ_j. This configuration typically gives an acceptance rate $\bar{\alpha}$ for each simulation level in the range of 0.2–0.4 [37]. This is supported by the numerical experiments performed with the examples in the next section, which leads to our recommendation for ABC-SubSim: *Adaptively choose the variance σ_j^2 of the jth intermediate level so that the monitored acceptance rate $\bar{\alpha} \in [0.2, 0.4]$ based on an initial chain sample of small length (e.g., 10 states).*

The choice of the conditional probability P_0 has a significant influence on the number of intermediate simulation levels required by the algorithm. The higher P_0 is, the higher the number of simulation levels employed by the algorithm to reach the specified tolerance ϵ for a fixed number of model evaluations (N) per simulation level. This necessarily increases the computational cost of the algorithm. At the same time, the smaller P_0 is, the lower the quality of the posterior approximation, that is, the larger the values of γ_j in (3.4). The choice of P_0 therefore requires a trade-off between computational efficiency and efficacy, in the sense of quality of the ABC posterior approximation.

To examine this fact, let us take a fixed total number of samples, i.e., $N_T = mN$, where m is the number of levels required to reach the target tolerance value ϵ, a tolerance for which $R_m^{(0)} \approx \mathrm{Var}\,[h(\theta)]$. The value of m depends on the choice of P_0. We can choose P_0 in an optimal way by minimizing the variance of the estimator $\bar{h}_m$ for the last simulation level:

$$\mathrm{Var}(\bar{h}_m) = \frac{R_m^{(0)}}{N_T/m}(1+\gamma_m) \propto m(1+\gamma_m). \tag{3.7}$$

Notice that γ_m also depends upon P_0, although it is not explicitly denoted, as we will show later in section 4 (Figure 2). In the original presentation of subset simulation in [1], $P_0 = 0.1$ was recommended, and more recently in [37], the range $0.1 \leqslant P_0 \leqslant 0.3$ was found to be near optimal after a rigorous sensitivity study of subset simulation, although the optimality there is related to the coefficient of variation of the failure probability estimate. The value $P_0 = 0.2$ for ABC-SubSim is also supported by the numerical experiments performed with the examples in the next section, where we minimize the variance in (3.7) as a function of P_0, which leads to the following recommendation: *For ABC-SubSim, set the conditional probability $P_0 = 0.2$.*

Finally, it is important to remark that an appropriate final tolerance ϵ may be difficult to specify a priori. For these cases, one recommendation is to select ϵ adaptively so that the posterior samples give a stable estimate $\bar{h}_m$ of $\mathbb{E}_{p_\epsilon(\theta|D_m)}[h(\theta)]$ (3.3), i.e., a further reduction in ϵ does not change $\bar{h}_m$ significantly.

3.2. Evidence computation by means of ABC-SubSim. In a modeling framework, different model classes can be formulated and hypothesized to idealize the experimental system, and each of them can be used to solve the probabilistic inverse problem in (1.1). If the modeler chooses a set of candidate model classes $\mathbf{M} = \{\mathcal{M}_k, k = 1, \ldots, N_M\}$, Bayesian model class assessment is a rigorous procedure to rank each candidate model class based on their probabilities conditional on data y [21, 7],

$$P(\mathcal{M}_k|y, \mathbf{M}) = \frac{p(y|\mathcal{M}_k)P(\mathcal{M}_k|\mathbf{M})}{\sum_{i=1}^{N_M} p(y|\mathcal{M}_i)P(\mathcal{M}_i|\mathbf{M})}, \tag{3.8}$$

where $P(\mathcal{M}_k|\mathbf{M})$ is the prior probability of each $\mathcal{M}_k$, that expresses the modeler's judgement on the initial relative plausibility of $\mathcal{M}_k$ within $\mathbf{M}$. The factor $p(y|\mathcal{M}_k)$,

which is called the *evidence* (or *marginal likelihood*) for the model class, expresses how likely the data y are according to the model class. The evidence $p(y|\mathcal{M}_k)$ is equal to the normalizing constant in establishing the posterior PDF in (1.1) for the model class:[3] $p(y|\mathcal{M}_k) = \int_{\boldsymbol{\Theta}} p(y|\theta, \mathcal{M}_k)p(\theta|\mathcal{M}_k)d\theta$.

When the likelihood is not available, the evidence $p(y|\mathcal{M}_k)$ is approximated using ABC by $P_\epsilon(y|\mathcal{M}_k)$, which depends upon ϵ, the summary statistic $\eta(\cdot)$ as well as the chosen metric ρ [28]. In terms of the notation in (3.1), the ABC evidence can be expressed as

$$P_\epsilon(y|\mathcal{M}_k) = P(D_m|\mathcal{M}_k) = \int_{\boldsymbol{\Theta}} P(D_m|\theta, x, \mathcal{M}_k)p(x|\theta, \mathcal{M}_k)p(\theta|\mathcal{M}_k)d\theta dx. \tag{3.9}$$

The evaluation of the last integral is the computationally expensive step in Bayesian model selection, especially when $\epsilon \to 0$ [22]. Observe that $P_\epsilon(y|\mathcal{M}_k)$ in (3.9) is expressed as a mathematical expectation that can be readily estimated as follows:

$$P_\epsilon(y|\mathcal{M}_k) \approx \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}_{D_m}\big(\theta^{(n)}, x^{(n)}\big), \tag{3.10}$$

where $(\theta^{(n)}, x^{(n)}) \sim p(x|\theta, \mathcal{M}_k)p(\theta|\mathcal{M}_k)$ are samples that can be drawn using the standard ABC algorithm (Algorithm 1), which in this setting is equivalent to the standard Monte Carlo method for evaluating integrals. The main drawback of this method arises when employing $\epsilon \to 0$, due to the well-known inefficiency of the standard ABC algorithm. Moreover, the quality of the approximation in (3.10) may be poor in this situation unless a huge amount of samples are employed because otherwise the Monte Carlo estimator has a large variance. Hence, several methods have emerged in the ABC literature to alleviate this difficulty, with the main drawback typically being the computational burden. See [12] for discussion of this topic.

ABC-SubSim algorithm provides a straightforward way to approximate the ABC evidence $P_\epsilon(y|\mathcal{M}_k)$ via the conditional probabilities involved in subset simulation:

$$P_\epsilon(y|\mathcal{M}_k) = P(D_m|\mathcal{M}_k) = P(D_1)\prod_{j=2}^{m} P(D_j|D_{j-1}) \approx P_0^m. \tag{3.11}$$

The last is an estimator for $P_\epsilon(y|\mathcal{M}_k)$ which is asymptotically unbiased with bias $\mathcal{O}(1/N)$. See [1, 37] for a detailed study of the quality of the estimators based on subset simulation where the approximation is studied in the context of the failure probability estimate (but notice that (3.11) and (2.2) are essentially the same). Of course, there are also approximation errors due to the ABC approximation that depend on the choice of $\epsilon, \eta(\cdot)$ and ρ [28]. Finally, once $P_\epsilon(y|\mathcal{M}_k)$ is calculated, it is substituted for $p(y|\mathcal{M}_k)$ in (3.8) to obtain $P_\epsilon(\mathcal{M}_k|y, \mathbf{M})$, the ABC estimate of the model class posterior probability. It is important to remark here that there are well-known limitations of the ABC approach to the model selection problem, typically attributable to the absence of sensible summary statistics that work across model classes, among others [12, 28]. Our objective here is to demonstrate that calculation of the ABC evidence is a simple by-product of ABC-SubSim, as given in (3.11).

4. Illustrative examples. In this section we illustrate the use of ABC-SubSim with two examples: (1) a moving average (MA) process of order $d = 2$, MA(2),

[3] The model parameter vector θ will, in general, be different for different model classes $\mathcal{M}_k$.

previously considered in [22], and (2) a single degree-of-freedom (SDOF) linear oscillator subject to white noise excitation, which is an application to a state-space model. Both examples are input-output type problems, in which we adopt the notation $y = [y_1, \ldots, y_l, \ldots, y_\ell]$ for the measured system output sequence of length ℓ. The objective of these examples is to illustrate the ability of our algorithm to be able to sample from the ABC posterior for small values of ϵ. In the MA(2) example, we take for the metric the quadratic distance between the $d = 2$ first autocovariances, as in [22]:

$$\rho\big(\eta(x), \eta(y)\big) = \sum_{q=1}^{d} (\tau_{y,q} - \tau_{x,q})^2. \tag{4.1}$$

In the last equation, the terms $\tau_{y,q}$ and $\tau_{x,q}$ are the autocovariances of y and x, respectively, which are used as summary statistics. They are obtained as $\tau_{y,q} = \sum_{k=q+1}^{\ell} y_k y_{k-q}$ and $\tau_{x,q} = \sum_{k=q+1}^{\ell} x_k x_{k-q}$, respectively. The Euclidean distance of x from y is considered as the metric for the oscillator example:

$$\rho(x, y) = \left[\sum_{l=1}^{\ell} (y_l - x_l)^2 \right]^{1/2}. \tag{4.2}$$

To evaluate the quality of the posterior, we study the variance of the mean estimator of a quantity of interest $h : \theta \in \mathbf{\Theta} \to \mathbb{R}$, defined as follows (see section 3.1.2):

$$h(\theta) = \sum_{i=1}^{d} \big(\theta_i\big)^2 = \|\theta\|_2^2. \tag{4.3}$$

4.1. Example 1: MA model. Consider a MA(2) stochastic process with $x_l, l = 1, \ldots, \ell$, the stochastic variable defined by

$$x_l = e_l + \sum_{i=1}^{d} \theta_i e_{l-i} \tag{4.4}$$

with $d = 2$, $\ell = 100$ or 1000. In the last equation, e is an independently and identically distributed (i.i.d.) sequence of standard Gaussian distributions $\mathcal{N}(0, 1)$: $e = [e_{-d+1}, \ldots, e_0, e_1 \ldots e_l, \ldots, e_\ell]$ and $x = [x_1, \ldots, x_l, \ldots, x_\ell]$. To avoid unnecessary difficulties, a standard identifiability condition is imposed on this model [22], namely, that the roots of the polynomial $\mathrm{D}(\xi) = 1 - \sum_{i=1}^{d} \theta_i \xi^i$ are outside the unit circle in the complex plane. In our case of $d = 2$, this condition is fulfilled when the region $\mathbf{\Theta}$ is defined as all (θ_1, θ_2) that satisfy

$$-2 < \theta_1 < 2; \theta_1 + \theta_2 > -1; \theta_1 - \theta_2 < 1.$$

The prior is taken as a uniform distribution over $\mathbf{\Theta}$.

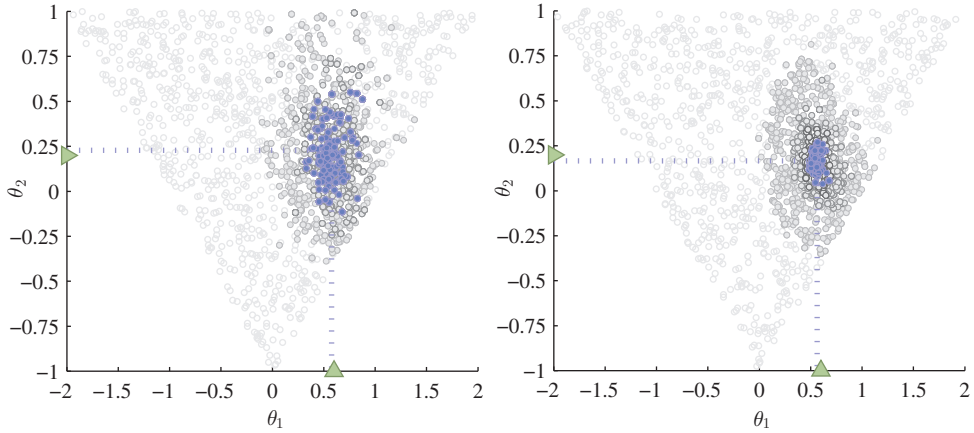
Note that, in principle, this example does not need ABC methods as the likelihood is a multidimensional Gaussian with zero mean and a covariance matrix of order ℓ that depends on (θ_1, θ_2), but its evaluation requires a considerable computational effort when ℓ is large [23]. This example was also used to illustrate the ABC method in [22] where it was found that the performance is rather poor if the metric is the one in (4.2) which uses the "raw" data, but ABC gave satisfactory performance when the metric in (4.1) was used. For comparison with Figure 1 in [22], we also choose the latter here.

TABLE 2

Parameter configuration of ABC-SubSim algorithm for the MA(2) and SDOF linear oscillator examples. The information shown in the first and second rows correspond to the MA(2) example with $\ell = 100$ and $\ell = 1000$, respectively. The values shown from the fourth to the seventh column correspond to the optimal values for the proposal standard deviation per simulation level for both examples.

Model	Sample size	Cond. probability	Proposal std. deviation				Sim. levels
	(N)	(P_0)	(σ_1)	(σ_2)	(σ_3)	(σ_4)	(m)
MA(2) ($\ell = 100$)	$1000^{(*)}$	0.2	0.4	0.2	0.1	$--$	3
MA(2) ($\ell = 1000$)	$1000^{(*)}$	0.2	0.4	0.2	0.1	0.04	4
Oscillator	$2000^{(*)}$	0.2	0.35	0.1	0.05	0.001	4

(*): per simulation level



(a) $\ell = 100$ (b) $\ell = 1000$

FIG. 1. *ABC-SubSim output for the MA(2) model with* (a) $\ell = 100$ *and* (b) $\ell = 1000$. *Each subplot presents samples (circles) in the model parameter space* $\boldsymbol{\Theta}$, *where the latest final posterior samples are marked in dark blue circles. The coordinates of the mean estimate of the latest posterior are represented by the blue dotted line. The green triangles are the coordinates of* θ_{true}. *To reveal the uncertainty reduction, the intermediate posterior samples are superimposed in increasing gray tones. Gray rings correspond to prior samples.*

We use synthetic data for y by generating it from (4.4) considering $\theta_{true} = (0.6, 0.2)$. The chosen values of the control parameters for ABC-SubSim are shown in Table 2. The ABC-SubSim results are presented in Figure 1, which shows that the mean estimate of the "approximate" posterior samples at each level is close to θ_{true} for both $\ell = 100$ and $\ell = 1000$ cases. Figure 1(a) shows the case $\ell = 100$ which can be compared with Figure 1 in [22]. In Figure 1(a), a total of 3000 samples were used to generate 1000 samples to represent the posterior, whereas in [22], 1,000,000 samples were used to generate 1000 approximate posterior samples using the standard ABC algorithm that we called Algorithm 1. The ABC-SubSim posterior samples give a more compact set that is better aligned with the exact posterior contours given in Figure 1 of [22]. Figure 1(b) shows that for the case $\ell = 1000$, ABC-SubSim used 4000 samples to generate 1000 samples representing the much more compact posterior that corresponds to ten times more data.

A preliminarily sensitivity study was done to corroborate the choice of the algorithm control parameters described in section 3.1.2, and the results are shown in Figure 2. As described in section 3.1.2, the optimal value of P_0 is the one that min-

imizes $m(1+\gamma_m)$ for fixed tolerance ϵ. As an exercise, we consider $\epsilon = 1.12 \cdot 10^4$ as the final tolerance.[4] The results in Figure 2 show that $P_0 = 0.2$ is optimal since then $m(1+\gamma_m) = 3(1+2.8) = 11.4$; whereas for $P_0 = 0.5$ and $P_0 = 0.1$, it is $7(1+0.86) = 13.1$ and $2(1+5.1) = 12.2$, respectively. These results are consistent with those for rare event simulation in [37]. Observe also that the optimal variance σ_j^2 for the Gaussian proposal PDF at the jth level that minimizes γ_j occurs when the acceptance rate $\bar{\alpha}_j$ in MMA lies in the range 0.2–0.4, which is also consistent with that found in [37] (except for the case of very low acceptance rate where the process is mostly controlled by the noise).

4.2. Example 2: Linear oscillator. Consider the case of a SDOF oscillator subject to white noise excitation as follows:

$$m\ddot{\xi} + c\dot{\xi} + k\xi = f(t), \tag{4.5}$$

where $\xi = \xi(t) \in \mathbb{R}$ $[m]$, m $[Kg]$, k $[N/m]$, and c $[N \cdot s/m]$ are the displacement, mass, stiffness, and damping coefficient, respectively. To construct synthetic input, a discrete-time history of input force f $[N]$ modeled by Gaussian white noise with spectral intensity $S_f = 0.0048\,[N^2 \cdot s]$ is used. The time step used to generate the input data is $0.01\,[s]$, which gives an actual value for the variance of the discrete input force $\sigma_f^2 = 3\,[N]$ [18, 36].

The probability model that gives the likelihood function of this example is Gaussian, and so it can be written explicitly although its evaluation requires the computation of a high-dimensional matrix inverse [35]. Repeated evaluations of the likelihood function for thousands of times in a simulation-based inference process is computationally prohibitive for large-size datasets. However, it is easy to simulate datasets from this model after some trivial manipulations of (4.5) [35]. Therefore, this example is particularly suited for the use of ABC methods.

The mechanical system is assumed to have known mass $m = 3$ $[Kg]$ and known input force giving the excitation. For the state-space simulation, denote the *state vector* by $s(t) = [\xi(t), \dot{\xi}(t)]^T$. Equation (4.5) can be rewritten in state-space form as follows:

$$\dot{s}(t) = A_c s(t) + B_c f(t), \tag{4.6}$$

where $A_c \in \mathbb{R}^{2\times 2}$, $B_c \in \mathbb{R}^{2\times 1}$ are obtained as

$$A_c = \begin{pmatrix} 0 & 1 \\ -m^{-1}k & -m^{-1}c \end{pmatrix}, \qquad B_c = \begin{pmatrix} 0 \\ m^{-1} \end{pmatrix}. \tag{4.7}$$

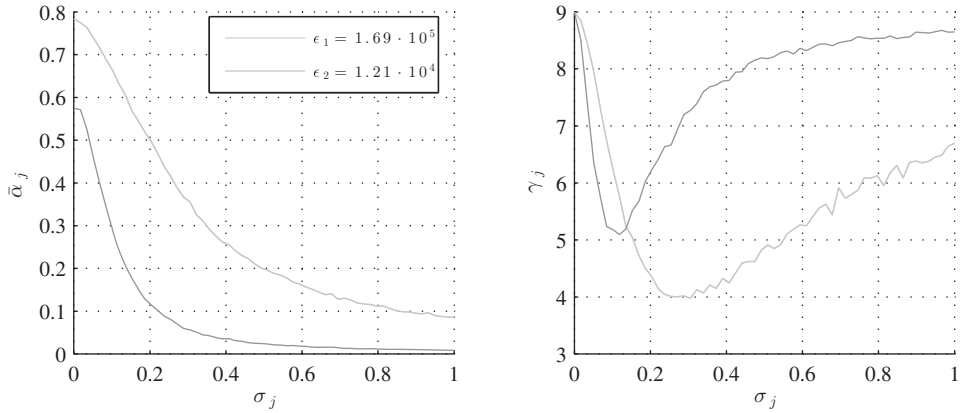
By approximating the excitation as constant within any interval, i.e., $f(l\triangle t + \tau) = f(l\triangle t)$ $\forall \tau \in [0, \triangle t)$, (4.6) can be discretized to a difference equation: $\forall l \geqslant 1$,

$$s_l = A s_{l-1} + B f_{l-1} \tag{4.8}$$

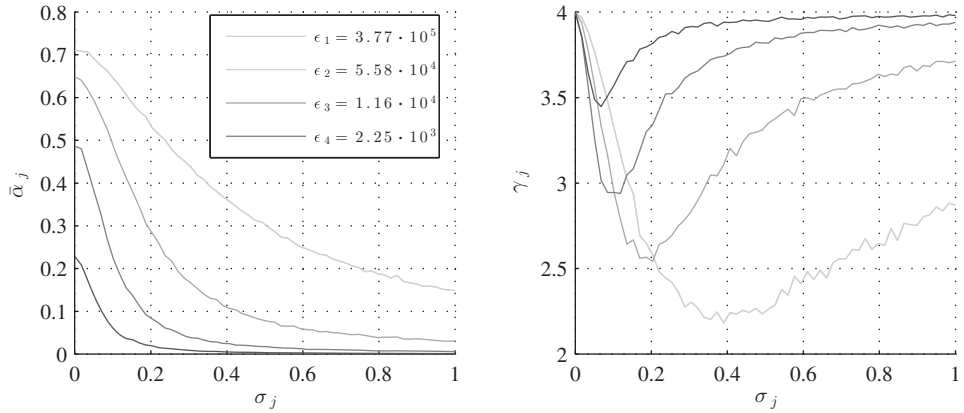
with $s_l \equiv s(l\triangle t)$, $f_l \equiv f(l\triangle t)$, $l = 0, 1, 2, \ldots, \ell$, and A and B are matrices given by

$$\begin{aligned} A &= e^{(A_c \triangle t)} \\ B &= A_c^{-1}\,(A - I_2)\,B_c, \end{aligned} \tag{4.9}$$

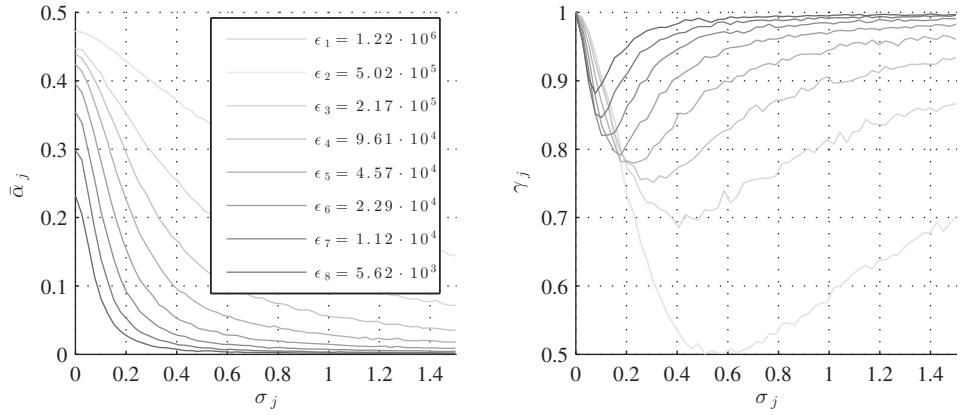
[4] It is unlikely that one or more values from the ϵ-sequence obtained using different P_0 values coincide exactly. Hence, the nearest value to the final tolerance is considered for this exercise.



(a) $P_0 = 0.1$



(b) $P_0 = 0.2$



(c) $P_0 = 0.5$

FIG. 2. *Sensitivity study of the acceptance rate $\bar{\alpha}_j$ and autocorrelation factor γ_j in relation to different values of the standard deviation σ_j for the MA(2) model with $\ell = 1000$ and for different values of $P_0 = 0.1(\text{a}), 0.2(\text{b}),$ and $0.5(\text{c})$. $N = 1000$ samples are employed per simulation level. Darker curves correspond to higher simulation levels. The tolerance values are indicated. The numerical values of each plot are obtained considering the mean of 50 independent runs of the algorithm.*

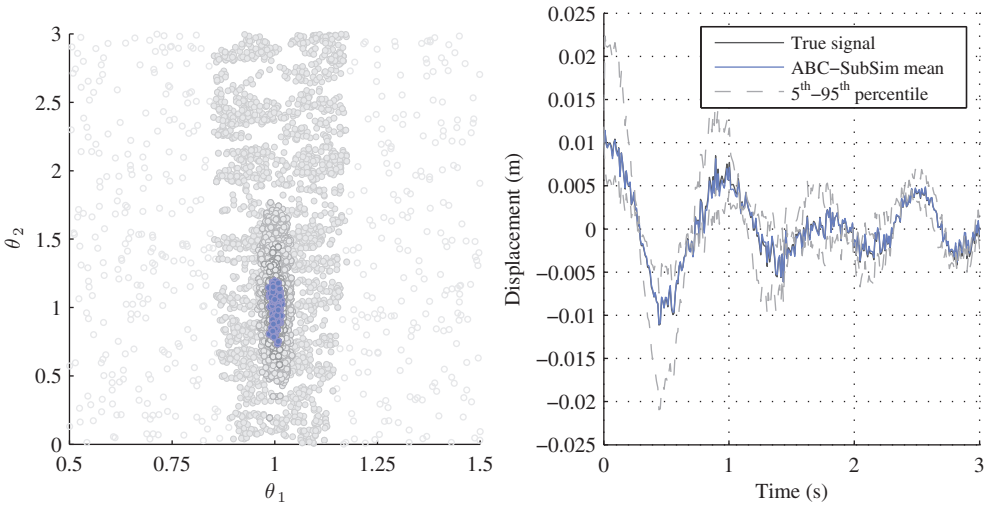


FIG. 3. *Results of the inference for the oscillator model for a duration of $t = 3$ seconds. Left: scatter plot of posterior samples of θ for intermediate levels and the final level (in blue). The horizontal and vertical scale are normalized by a factor of 4π and 0.4π, respectively. Right: synthetic signal response of the oscillator, together with the mean estimate of the ABC-SubSim approximation and two percentiles.*

where I_2 is the identity matrix of order 2. The use of discrete-time input and output data here is typical of the electronically collected data available from modern instrumentation on mechanical or structural systems.

We adopt $\theta = \{k, c\}$ as unknown model parameters and denote by y_l and x_l the vectors consisting of the actual and predicted response measurements at each $\triangle t$. Samples of x_l for a given input force time history $\{f_l\}$ and θ can be readily generated by the underlying state-space model:

$$s_l = As_{l-1} + Bf_{l-1} + e_l, \tag{4.10a}$$

$$x_l = [1, 0]s_l + e'_l, \tag{4.10b}$$

where e_l and e'_l are error terms to account for model prediction error and measurement noise, respectively. Since in reality these errors would be unknown, we use the principle of maximum information entropy [19, 20, 6] to choose e_l and e'_l as i.i.d. Gaussian variables, $e_l \sim \mathcal{N}(0, \sigma_e^2 I_2)$, $e'_l \sim \mathcal{N}(0, \sigma_{e'}^2)$ and so they can be readily sampled. For simplicity, we adopt $\sigma_e^2 = 10^{-2}$ and $\sigma_{e'}^2 = 10^{-6}$, taking them as known. We call $y = \{y_1, \ldots, y_l, \ldots, y_\ell\}$ the batch dataset collected during a total period of time $t = \ell\triangle t$, starting from known initial conditions $s_0 = [0.01, 0.03]^T$ (units expressed in $[m]$ and $[m/s]$, respectively). In this example, the noisy measurements y_l are synthetically generated from (4.10) for the given input force history and for model parameters $\theta_{true} = \{k = 4\pi, c = 0.4\pi\}$. We also adopt a sampling rate for the resulting output signal of 100 [Hz] ($\triangle t = 0.01[s]$) during a sampling period of $t = 3[s]$, and hence $\ell = 300$. We choose a uniform prior over the parameter space $\boldsymbol{\Theta}$ defined by the region $0 < \theta_i \leqslant 3$; $i \in \{1, 2\}$. Table 2 provides the information for the algorithm configuration.

The results shown in Figure 3 are very satisfactory in the sense that ABC-SubSim can reconstruct the true signal with high precision with only a moderate computational cost. The posterior samples show that in Bayesian updating using noisy input-

TABLE 3
Set of tolerance values used for comparing the sequential ABC algorithms established using ABC-SubSim with $P_0 = 0.5$.

Model	ϵ_1	ϵ_2	ϵ_3	ϵ_4	ϵ_5	ϵ_6	ϵ_7	ϵ_8	ϵ_9	ϵ_{10}
MA(2) ($\times 10^4$)	122	50.2	21.7	9.61	4.57	2.29	1.12	0.56	0.28	0.14
Oscillator	0.0117	0.0099	0.0082	0.0054	0.0040	0.0030	0.0024	0.0020	0.0018	0.0016

output data, the stiffness parameter $k = 4\pi\theta_1$ is identified with much less uncertainty than the damping parameter $c = 0.4\pi\theta_2$. The normalized mean value over the set of posterior samples corresponding to the smallest value of ϵ is $\bar{\theta} = (1.00, 1.03)$, which is very close to the normalized true value $\theta_{true} = (1.0, 1.0)$. (Even if the exact likelihood was used, we would not expect $\bar{\theta} = \theta_{true}$ because of the noise in the synthetic data y.)

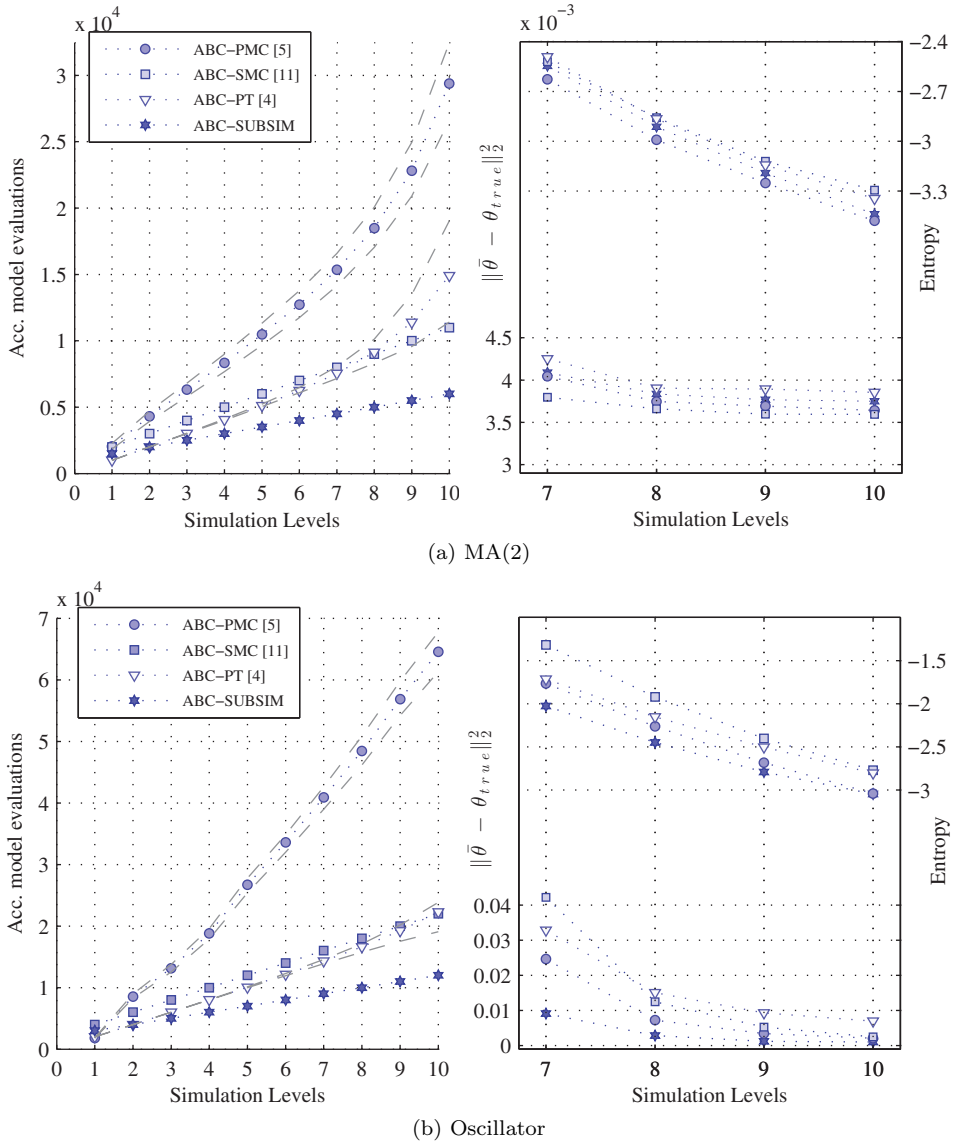
5. Discussion.

5.1. Comparison with recent sequential ABC algorithms. In this section, ABC-SubSim is compared with a selection of recent versions of sequential ABC algorithms: ABC-SMC [11], ABC-PMC [5], and ABC-PT [4], which are listed in Table 1. The same number of evaluations per simulation level is adopted for all algorithms, corresponding to 1000 and 2000 for the MA(2) and SDOF model, respectively. We set the sequence of tolerance levels obtained by ABC-SubSim using $P_0 = 0.5$ for the rest of the algorithms (see Table 3). This was done because the recommended near-optimal value of $P_0 = 0.2$ (see section 3.1.2) for ABC-SubSim produced a sequence of ϵ values that decreased too quickly for ABC-PMC and ABC-SMC to work properly. We note that this nonoptimal choice of P_0 for ABC-SubSim and the use of its ϵ-sequence provide considerable help for the competing algorithms. The proposal PDFs are assumed to be Gaussian for all of the algorithms.

The results shown in Figure 4 are evaluated over the intermediate posterior samples for each simulation level and were obtained considering the mean of 100 independent runs of the algorithms, a large enough number of runs to ensure the convergence of the mean. In this example, we focus on the number of model evaluations together with the quality of the posterior. The left side of Figure 4 shows the accumulated amount of model evaluations employed by each of the competing algorithms. Note that each algorithm requires the evaluation of auxiliary calculations, like those for the evaluation of particle weights, transition kernel steps, etc. However, this cost is negligible because the vast proportion of computational time in ABC is spent on simulating the model repeatedly. The number of model evaluations for ABC-PMC and ABC-PT is variable for each algorithm run, so in both cases we present the mean (labelled dotted lines) and a 95% band (dashed lines). In contrast, ABC-SubSim and also ABC-SMC make a fixed number of model evaluations at each simulation level. Observe that the computational saving is markedly high when comparing with ABC-PMC.

Regarding the quality of the posterior, we consider two measures: (a) the sample mean of the quadratic error between $\bar{\theta}$ and θ_{true}, i.e., $\|\bar{\theta}_j - \theta_{true}\|_2^2$, as an accuracy measure, and (b) the differential entropy[5] of the final posterior, by calculating $1/2 \ln |(2\pi e)^d \det [cov(\theta_j)]|$, as a measure quantifying the posterior uncertainty of the model parameters. The results are shown on the right side of Figure 4. Only the last four simulation levels are presented for simplicity and clearness.

[5]This expression for the differential entropy is actually an upper-bound approximation to the actual differential entropy, where the exactness is achieved when the posterior PDF is Gaussian.



(a) MA(2)

(b) Oscillator

FIG. 4. *Left: Accumulated model evaluations per simulation level for* (a) *MA*(2), (b) *oscillator. Right: Differential entropy (right-side of the y-label) of the intermediate posterior samples and mean quadratic error between* $\bar{\theta}$ *and* θ_{true} *(left-side of the y-label). Both measures are evaluated for the last four intermediate simulation levels:* $\epsilon_j, j = 7, 8, 9, 10$. *To be equivalent to ABC-SubSim, we consider for the implementation of the ABC-SMC algorithm a percentage of alive particles* $\alpha = 0.5$ *and* $M = 1$ *(see the details in* [11]*).*

This comparison shows that ABC-SubSim gives the same, or better, quality than the rest of the ABC algorithms to draw ABC posterior samples when ϵ is small enough, even though it used a smaller number of model evaluations.

5.2. Evidence calculation. In this section we show how the ABC-SubSim algorithm can be applied to estimate the ABC evidence by taking advantage of the improvements in parameter space exploration introduced by subset simulation. Table 4 shows the estimated values of the ABC evidence obtained with the ABC-SubSim

TABLE 4

Results of the estimation of the ABC evidence $P_{\epsilon_j}(\mathcal{D}_j|\mathcal{M})$ for the MA(2) and oscillator examples when using 4 different tolerance values $\epsilon_j, j = 1, \ldots, 4$, which are produced by the ABC-SubSim algorithm with $P_0 = 0.2$. The Standard ABC algorithm employing 200,000 samples is also used to estimate $P_{\epsilon_j}(\mathcal{D}_j|\mathcal{M})$ as in (3.10).

Example 1: MA(2)			Example 2: Oscillator		
	SubSim	Standard ABC		SubSim	Standard ABC
$(\epsilon_1 = 3.77 \cdot 10^5)$	0.2	0.2070	$(\epsilon_1 = 0.0053)$	0.2	0.2038
$(\epsilon_2 = 5.58 \cdot 10^4)$	0.04	0.0412	$(\epsilon_2 = 0.0023)$	0.04	0.0397
$(\epsilon_3 = 1.16 \cdot 10^4)$	0.008	0.0078	$(\epsilon_3 = 0.0016)$	0.008	0.0079
$(\epsilon_4 = 2.25 \cdot 10^3)$	0.0016	0.0017	$(\epsilon_4 = 0.0014)$	0.0016	0.0016

algorithm ($P_0 = 0.2$), which are computed using a total number of samples per simulation level N equal to 1000 and 2000 for MA(2) and SDOF model, respectively. For each value of ϵ_j chosen adaptively by ABC-SubSim as described in section 3.1.1, we also calculate the ABC evidence using the approximation in (3.10) with $N = 200{,}000$ samples per ϵ value for the standard ABC algorithm (a large enough amount of samples for the approximation in (3.10) to be sufficiently accurate). It is seen in both examples that the results obtained by ABC-SubSim and standard ABC agree well.

These results suggest that if the well-known difficulties of the ABC model choice problem can be adequately resolved, high efficiency can be obtained by employing the ABC-SubSim algorithm for the ABC evidence computation.

6. Conclusions. A new ABC algorithm based on MCMC has been presented and discussed in this paper. This algorithm combines the principles of ABC with a highly efficient rare-event sampler, subset simulation, which draws conditional samples from a nested sequence of subdomains defined in an adaptive and automatic manner. We demonstrate the computational efficiency that can be gained with ABC-SubSim by two different examples that illustrate some of the challenges in real-world applications of ABC. The main conclusions of this work are as follows:

(i) By its construction, ABC-SubSim avoids the difficulties of ABC-MCMC algorithm in initializing the chain, as no burn-in is required.

(ii) In comparison with other recent sequential ABC algorithms, ABC-SubSim requires a smaller number of model evaluations per simulation level to maintain the same quality of the posterior as the other algorithms.

(iii) Together with ABC-SMC from [11], ABC-SubSim does not require the specification of a sequence of tolerance levels, which avoids tedious preliminary calibrations.

(iv) ABC-SubSim allows a straightforward way to obtain an estimate of the ABC evidence used for model class assessment.

Acknowledgment. The first two authors would like to thank the California Institute of Technology (Caltech) which kindly hosted them during the course of this work.

REFERENCES

[1] S.K. AU AND J.L. BECK, *Estimation of small failure probabilities in high dimensions by subset simulation*, Probabilistic Engrg. Mech., 16 (2001), pp. 263–277.

[2] S.K. AU AND J.L. BECK, *Subset simulation and its application to seismic risk based on dynamic analysis*, J. Eng. Mech., 129 (2003), pp. 901–917.

[3] S.K. AU, J. CHING, AND J.L. BECK, *Application of subset simulation methods to reliability benchmark problems*, Struct. Safety, 29 (2007), pp. 183–193.

[4] M. Baragatti, A. Grimaud, and D. Pommeret, *Likelihood-free parallel tempering*, Statist. Comput., 23 (2013), pp. 535–549.
[5] M.A. Beaumont, J.M. Cornuet, J.M. Marin, and C.P. Robert, *Adaptive approximate Bayesian computation*, Biometrika, 96 (2009), pp. 983–990.
[6] J.L. Beck, *Bayesian system identification based on probability logic*, Struct. Control Health Monitoring, 17 (2010), pp. 825–847.
[7] J.L. Beck and K.V. Yuen, *Model selection using response measurements: Bayesian probabilistic approach*, J. Eng. Mech., 130 (2004), p. 192.
[8] P. Bortot, S.G. Coles, and S.A. Sisson, *Inference for stereological extremes*, J. Amer. Statist. Assoc., 102 (2007), pp. 84–92.
[9] J. Ching, S.K. Au, and J.L. Beck, *Reliability estimation of dynamical systems subject to stochastic excitation using subset simulation with splitting*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1557–1579.
[10] P. Del Moral, A. Doucet, and A. Jasra, *Sequential Monte Carlo samplers*, J. R. Stat. Soc. Ser. B Stat. Methodol., 68 (2006), pp. 411–436.
[11] P. Del Moral, A. Doucet, and A. Jasra, *An adaptive sequential Monte Carlo method for approximate Bayesian computation*, Statist. Comput., 22 (2012), pp. 1009–1020.
[12] X. Didelot, R.G. Everitt, A.M. Johansen, and D.J. Lawson, *Likelihood free estimation of model evidence*, Bayesian Anal., 6 (2011), pp. 49–76.
[13] C.C. Drovandi and A.N. Pettitt, *Estimation of parameters for macroparasite population evolution using approximate Bayesian computation*, Biometrics, 67 (2011), pp. 225–233.
[14] P. Fearnhead and D. Prangle, *Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation*, J. R. Stat. Soc. Ser. B Stat. Methodol., 74 (2012), pp. 419–474.
[15] A. Gelman, G. Roberts, and W. Gilks, *Efficient Metropolis jumping rules*, Bayesian Statist., 5 (1996), pp. 599–608.
[16] W.R. Gilks, *Markov Chain Monte Carlo*, in Encyclopedia of Biostatistics, Wiley Online Library, 2005.
[17] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London, 1996.
[18] M.H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley, New York, 2009.
[19] E.T. Jaynes, *Information theory and statistical mechanics*, Phys. Rev., 106 (1957), pp. 620–630.
[20] E.T. Jaynes, *Probability Theory: The Logic of Science*, G.L. Bretthorst, ed., Cambridge University Press, Cambridge, 2003.
[21] D.J.C. MacKay, *Bayesian interpolation*, Neural Comput., 4 (1992), pp. 415–447.
[22] J.M. Marin, P. Pudlo, C.P. Robert, and R.J. Ryder, *Approximate Bayesian computational methods*, Stat. Comput., 22 (2012), pp. 1167–1180.
[23] J.M. Marin and C.P. Robert, *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*, Springer-Verlag, New York, 2007.
[24] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré, *Markov chain Monte Carlo without likelihoods*, Proc. Natl. Acad. Sci. USA, 100 (2003), pp. 15324–15328.
[25] R.M. Neal, *Probabilistic Inference Using Markov Chain Monte Carlo Methods*, Technical report CRG TR 93 1, Department of Computer Science, University of Toronto, Toronto, 1993.
[26] J.K. Pritchard, M.T. Seielstad, A. Perez-Lezaun, and M.W. Feldman, *Population growth of human Y chromosomes: A study of Y chromosome microsatellites*, Molecular Biology and Evolution, 16 (1999), pp. 1791–1798.
[27] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd ed., Springer-Verlag, New York, 2004.
[28] C.P. Robert, J.M. Cornuet, J.M. Marin, and N.S. Pillai, *Lack of confidence in approximate Bayesian computation model choice*, Proc. Natl. Acad. Sci. USA, 108 (2011), pp. 15112–15117.
[29] G.O. Roberts and J.S. Rosenthal, *Optimal scaling for various Metropolis–Hastings algorithms*, Statist. Sci., 16 (2001), pp. 351–367.
[30] S.A. Sisson and Y. Fan, *Likelihood-free Markov chain Monte Carlo*, in Handbook of Markov Chain Monte Carlo, Chapman and Hall/CRC Press, Boca Raton, FL, 2011, pp. 319–341.
[31] S.A. Sisson, Y. Fan, and M.M. Tanaka, *Sequential Monte Carlo without likelihoods*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 1760–1765.
[32] S.A. Sisson, Y. Fan, and M.M. Tanaka, *A Note on Backward Kernel Choice for Sequential Monte Carlo without Likelihoods*, Technical report, University of New South Wales, Kensington, Australia, 2009.

[33] S. Tavare, D.J. Balding, R.C. Griffiths, and P. Donnelly, *Inferring coalescence times from DNA sequence data*, Genetics, 145 (1997), p. 505.

[34] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M.P.H. Stumpf, *Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems*, J. R. Soc. Interface, 6 (2009), pp. 187–202.

[35] K.V. Yuen, *Bayesian Methods for Structural Dynamics and Civil Engineering*, John Wiley, New York, 2010.

[36] K.V. Yuen and J.L. Beck, *Updating properties of nonlinear dynamical systems with uncertain input*, J. Eng. Mech., 129 (2003), pp. 9–20.

[37] K.M. Zuev, J.L. Beck, S.K. Au, and L.S. Katafygiotis, *Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions*, Computers & Structures, 93 (2011), pp. 283–296.